REMARKS

This Amendment is responsive to the Office Action dated March 18, 2004. All rejections and objections of the Examiner are respectfully traversed. Reconsideration is respectfully requested.

At paragraphs 3-5 of the Office Action, the Examiner requested that application serial numbers be provided in the list of co-pending applications in the Specification. Amendments to the Specification herein are believed to satisfy all requirements of the Examiner in this regard.

At paragraphs 6-8 of the Office Action, the Examiner rejected claims 18 and 30 for indefiniteness under 35 U.S.C. 112, second paragraph. Amendments to the claims herein are respectfully believed to satisfy all requirements of the Examiner in this regard.

At paragraphs 9-27, the Examiner rejected claims 1-11, 13-23, and 25-35 for obviousness under 35 U.S.C. 103(a), citing the combination of United States Patent number 6,539,435 of Bolmarcich et al. ("Bolmarcich et al.") and United States Patent number 5,911,066 of Williams et al. ("Williams et al."). Applicants respectfully traverse this rejection.

Bolmarcich et al. disclose a system for establishing a communication connection between two programs, each running on multiple processors of a distributed or shared memory parallel computer, or on multiple computers in a cluster of workstations or a set of network connected workstations. Bolmarcich et al. specifically teaches establishing communication between two parallel programs on a session basis. Once communication is established between the two parallel programs using the Bolmarcich et al. system, then all tasks of either program are able to freely communicate with each other, using a communication mechanism such as message passing. Bolmarcich et al. go on to describe a task of an active client program making a function

call requesting a connection to a passive server program. The passive program in the Bolmarcich et al. system can be identified by a well known name, or by a name discovered by querying a central manager. The client program manager of Bolmarcich et al. sends a request to connect to the passive server program manager. To locate the server program manager, the Bolmarcich et al. client inquires of a central manager regarding the location of the server program manager. When it receives the connection request, the program manager detects that all server tasks have updated their routing tables, either by polling all the semaphores, or by receiving update messages from all the server tasks, and informs the client program manager that the server side of the connection is complete. The Bolmarcich et al. client program manager then updates client routing tables and releases all the client tasks. Receipt of a message from a client task completes an implicit notification of the server tasks in Bolmarcich et al. that the client tasks are now ready to receive messages from them.

Williams et al. disclose a uniform data transfer mechanism is used by each computer program performing data transfer, providing a communication mechanism and a flexible and robust interface to support existing and future data transfer applications. The Williams et al. mechanism is described as potentially reused by multiple computer programs that need not include their own code to handle uniform data transfer. The Williams et al. system provides interfaces for facilitating uniform data transfer, including a dynamically linked library having interface groups with methods useful to a software object that transfers data. Upon receipt of a pointer to the appropriate interface, the Williams et al. client may use the uniform data transfer system. The Williams et al. disclosure includes a data structure describing the format in which data being transferred is conveyed, as well as the device for which the data being transferred is destined, and that allows a client to request data from an object.

Nowhere in the combination of Bolmarcich et al. and Williams et al. is there disclosed or suggested any method or system for establishing communication between a first application and a second application, the second application executing on a platform, that operates in whole or in part by:

> forwarding a notify message to the second application, receipt of the notify message by the second application causing the second application to ascertain path data for establishing a path between the first application and the second application, the notify message including a unique identifier to name the path, the unique identifier associated with a specific type of information to be transferred on the path, ***wherein the forwarding is from the first application to the second application, and wherein the forwarding is responsive to the first application being added to the platform***; . . . (emphasis added)

as in the present independent claim 1. Independent claims 13 and 25 include analogous features. In contrast, Bolmarcich et al. describes a system in which a program manager of a client sends a message *to the program manager of the server program*. The program manager of the server program in Bolmarcich et al. then sends a message *to the task manager at each of the server program's tasks*. The task manager of Bolmarcich et al. is *a separate process at each of the nodes*. The Bolmarcich et al. server program task subsequently examines a message queue for messages *from the task manager*, including *a message to update the routing table to add a newly connected client program*. See Bolmarcich et al. beginning at line 37 of column 7, through column 10, line 14. In contrast to the above cited features of the present independent claims 1, 13 and 25, none of the messaging described in Bolmarcich et al. is between a first application and a second application to establish a communication path between the first application and the second application, and none of the messaging in Bolmarcich et al. is responsive to an application being added to a platform. Similarly, Williams et al. includes no hint or suggestion of providing messages of any kind in response to an application being added to a platform.

Neither Bolmarcich et al. nor Williams et al. are addressed to the problem of adding new application programs to a platform.

For the above reasons, Applicants respectfully urge that the combination of Bolmarcich et al. and Williams et al. does not disclose or suggest all the features of the present independent claims 1, 13 and 25. Accordingly, the combination of Bolmarcich et al. and Williams et al. does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to the independent claims 1, 13 and 25. As to dependent claims 2-11, 14-23, and 26-35, they each depend from claims 1, 13 and 25, and are believed to be patentable over the combination of Bolmarcich et al. and Williams et al. for at least the same reasons.

In paragraphs 28-30 the Examiner rejected claims 12, 24 and 36 for obviousness under 35 U.S.C. 103, again citing Bolmarcich et al. and Williams et al. and additionally citing United States patent number 5,539,886 of Aldred et al. ("Aldred et al."). Applicants respectfully traverse this rejection.

Aldred et al. disclose a programmable workstation for collaborative working in a network including a conventional operating system and network control layer for controlling physical routing of data between nodes. A collaborative application subsystem in Aldred et al. interfaces with application programs, and is responsive to a predetermined call from a collaboration call manager to establish the call manager at a node to handle incoming events which are not specific to any application program instances at the node. Aldred et al. additionally teach port event handler functionality for processing commands associated with specific, associated ports. The relevant disclosures of Bolmarcich et al. and Williams et al. are discussed above with regard to the rejection in paragraphs 9-27 of the Office Action.

Nowhere in the combination of Bolmarcich et al., Williams et al. and Aldred et al. is there disclosed or suggested any method or system for establishing communication between a first application and a second application, the second application executing on a platform, that operates in whole or in part by:

> forwarding a notify message to the second application, receipt of the notify message by the second application causing the second application to ascertain path data for establishing a path between the first application and the second application, the notify message including a unique identifier to name the path, the unique identifier associated with a specific type of information to be transferred on the path, ***wherein the forwarding is from the first application to the second application, and wherein the forwarding is responsive to the first application being added to the platform***; . . . (emphasis added)

as in the present independent claim 1. Independent claims 13 and 25 include analogous features. Like Bolmarcich et al. and Williams et al., Aldred et al. is not addressed to the problem of adding new application programs to a platform.

For the above reasons, Applicants respectfully urge that the combination of Bolmarcich et al., Williams et al. and Aldred et al. does not disclose or suggest all the features of the present independent claims 1, 13 and 25, from which claims 12, 24 and 36 depend. Accordingly, the combination of Bolmarcich et al., Williams et al. and Aldred et al. does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to the independent claims 1, 13 and 25. As to dependent claims 12, 24 and 36, they each depend from claims 1, 13 and 25, and are believed to be patentable over the combination of Bolmarcich et al., Williams et al. and Aldred et al. for at least the same reasons. Reconsideration of all pending claims is respectfully requested.

In view of the above, the claims are believed to be allowable, and Applicants respectfully request that the Examiner's rejections of claims 1-36 be withdrawn.

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone David A. Dagg, Applicants' Attorney at 978-264-6664 so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

August 16 2004
Date

David A. Dagg, Reg. No. 37,809
Attorney/Agent for Applicant(s)
Steubing McGuinness & Manaras LLP
125 Nagog Park Drive
Acton, MA 01720
(978) 264-6664

Docket No. 120-035